

MicroByte

Software for the 2650

FORTH

MicroByte FORTH V1.0 is an implementation of FORTH-79 for the 2650. It comprises a FORTH interpretive compiler, a simple editor written in FORTH, and complete source codes in machine readable form.

FORTH is an ideal language for implementation on microcomputers because it has a very small run time overhead for a language of its power and flexibility.

What is FORTH?

FORTH is a computer language originally created for applications in Observatories which has experienced a considerable increase in popularity recently, amongst mini and microcomputer enthusiasts. FORTH is an excellent language in which to write small and medium sized 'one off' programs because its enforced block structure helps ensure that programs will be logically correct. After an initial learning phase, programming in FORTH is very much faster than programming in other high level languages.

FORTH is a 'hackers' language. If you love to 'get into' your machine, then FORTH is for you!

Initially FORTH has an an alarming unfamiliarity, because of the use of Reverse Polish (postfix) Notation, both for arithmetic operations as well as the syntax of FORTH words. However once this is learnt FORTH enables programs to be written extremely rapidly with a minimum of bugs and development problems.

FORTH words can be compiled individually in the interactive mode to test them one at a time. Once each word has been tested, an entire program can be entered in source form with the FORTH editor (itself a FORTH program) and then compiled. Once a new word is compiled, it becomes a part of the FORTH language itself.

FORTH uses a stack to pass parameters between FORTH words. FORTH initially consists of about 200 predefined words. Writing a program in FORTH consists of defining new FORTH words, in terms of previously defined words, until one word defines the entire program.

Not only is the language itself 'extensible', by defining new words, but completely new data structures can be specified. Although FORTH comes supplied with only 16 bit integer, 8 bit character and simple string functions, other data types such as floating point numbers, arrays or any other data type you care to think of may be defined as part of FORTH.

Compiled FORTH code is compact. 8K of memory will support the 6K run-time system and also permit quite extensive FORTH programs to be written.

PO Box 274 Belconnen ACT 2616

Machine language routines may be incorporated into FORTH as words where the extra speed is required.

FORTH v1.0 is a subset of the FORTH-79 standard as defined by the Forth Standards Team in the document 'FORTH - 79', distributed by the FORTH Interest Group.

Features:

- Produces fast, compact, compiled programs.
- Ideal for small and medium sized programming tasks.
- Permits unlimited access to the machine level of the computer.
- Forces a coherent and block structured approach to programming.
- An expandable language.
- Easy to define new FORTH words or new data types or structures.
- Assembler routines may be incorporated into FORTH.

REQUIREMENTS

Memory	Scratch RAM	0440 - 07FF
	Program	0800 - 2500 (Nominal)
	Buffer and data stack	2500 - 3FFF

Hardware None required, however FORTH can use any devices available. Routines are provided for BINBUG Binary and ACOS cassette formats and MicroByte DOS.

COST

Cassette	(BINBUG Binary format) 300 Baud	\$ 40
	ACOS format	\$ 35
Disk	MicroByte DOS 5" 35 track	\$ 35

All versions are supplied with a manual and complete source codes in machine readable form.

Note: - Assembler source code requires ASSEMBLER V2.0 to assemble.

Not available in 110 Baud format.

POST AND PACKING

Surface mail within Australia	\$ 2
Airmail - Australia, Zone 1, and Zone 2	\$ 5
- Anywhere else	\$ 10

AVAILABILITY

Mail order sales MicroByte,
P.O. BOX 274,
Belconnen. 2616

April 1982